文章编号: 1001-0920 (2014) 01-0039-06 DOI: 10.13195/j.kzyjc.2012.1764

基于场景重构与报警聚合的网络取证分析技术

董晓梅[1], 赵 茜[1], 李晓华[1], 费雅洁[2]

(1. 东北大学 信息科学与工程学院，沈阳 110819；2. 沈阳工程学院 信息工程系，沈阳 110136)

摘 要: 提出一种包含报警标准化、去冗余、场景重构和报警聚合的网络取证分析方法. 通过去除失败攻击的报警, 减少了对证据分析的干扰. 在场景重构中, 通过反向关联, 减少了不必要的证据, 同时通过对孤立报警的补充, 保证了证据链的完整性. 在报警聚合中, 提出了聚合同一攻击步骤的不同报警的方法, 以抽象层和具体层两个层次重构入侵场景. 最后通过实验验证了所提出方法的有效性.

关键词: 网络取证；去冗余；场景重构；报警聚合

中图分类号: TP393 **文献标志码:** A

Network forensics based on scenario reconstruction and alert aggregation

DONG Xiao-mei[1], *ZHAO Qian*[1], *LI Xiao-hua*[1], *FEI Ya-jie*[2]

(1. College of Information Science and Engineering，Northeastern University，Shenyang 110819，China；2. Department of Information Engineering，Shenyang Institute of Engineering，Shenyang 110136，China. Correspondent：DONG Xiao-mei，E-mail：dongxiaomei@ise.neu.edu.cn)

Abstract: A network forensics research method is proposed, which includes alert standardization, alert redundancy reduction, scenario reconstruction and alert aggregation. The interference of failed attacks to the forensics process is reduced by removing the failed alert. In the process of scenario reconstruction, with the method of inversely association, the unnecessary evidence can be removed. Moreover, isolated alerts are supplemented to ensure the integrity of evidence chain. In the process of alert aggregation, the method of merging different detailed alerts of the same step is proposed. The intrusion scenarios at the abstract layer and the specific layer are reconstructed respectively. Finally, experiments verify the effectiveness of the proposed method.

Key words: network forensics；redundancy reduction；scenario reconstruction；alert aggregation

0 引 言

随着网络攻击水平的不断提高和攻击工具、攻击方法的日新月异, 利用计算机网络进行犯罪的行为越来越严重. 网络取证技术的研究对于打击网络犯罪具有重要意义.

由于入侵检测系统的准确性达不到100%, 目前已有的很多网络取证方法都难以解决因误报和漏报导致的取证结果不准确、不完整的问题. 本文提出了一种先进行因果关联后进行报警聚合的方法: 在场景重构中, 通过去冗余技术来减少攻击失败报警对场景重构过程的干扰, 提高了重构的准确性和效率; 采用因果关联的方法, 从时间约束、状态约束、参数约束3个方面对报警进行直接关联; 利用反向搜索来补充遗漏的报警事件, 去除误报警. 因此, 本文提出的方法可以在很大程度上消除漏报和误报对网络取证结果的影响. 在实验中, 采用DARPA 2000的入侵检测数据集, 并利用基于Snort网络入侵检测系统的报警信息, 对本文提出的方法进行了验证.

1 相关工作

网络证据就是正在网上传输的计算机证据, 其实质是网络数据流[1]. 网络证据的获取属于动态取证, 即在犯罪事件进行时或在证据数据的传输途中进行截获. 目前网络取证的方法主要有两种: 一种是基于蜜阱的方法, 另一种是通过分析入侵检测系统报警的方法.

蜜阱主要包括蜜罐和蜜网等以诱骗技术为核心的网络安全技术[2]. 由于蜜阱是一种诱骗系统, 提供

收稿日期: 2012-11-26；修回日期: 2013-01-24.

基金项目: 教育部中央高校基本科研业务费基金项目(N100404005).

作者简介: 董晓梅(1970−), 女, 副教授, 从事网络安全等研究；赵茜(1987−), 女, 硕士生, 从事网络安全的研究.

的是一些非真实性的信息, 这些信息能否作为证据使用还值得商榷.

针对入侵检测系统的报警进行分析, 可以通过已经得到的证据来还原入侵者采取的一系列攻击步骤, 目前已有的网络取证方法大多是这种方法.

文献 [3] 提出了一种基于属性相似度的算法, 它根据属性间的相似度来对报警进行分组和关联. 另外, 文献 [4-6] 基于攻击的前因后果的重构方法, 利用入侵步骤中的因果关系来关联相应的报警; 文献 [7] 基于统计的方法, 利用统计学算法来发现报警间的因果关系, 并据此建立预测模型来关联新的报警. 还有一些基于已知攻击场景的方法, 例如: 基于网络攻击图的方法[8]、基于预定义的方法[9]、基于案例学习的方法[10]、基于策略因果网的方法[11]和基于数据挖掘的方法[12]等. 文献 [13] 提出了一种协同取证方法, 综合使用报警关联、贝叶斯网络和概率函数依赖理论进行报警的分析.

由于入侵检测系统的准确性难以保证, 也限制了以上网络取证方法的准确性, 如何消除误报和漏报对取证准确性的影响, 是这些方法面临的一个难题. 同时, 入侵检测依赖于捕获的网络数据片段, 所获得的证据很可能不够完整. 本文提出的方法中, 通过报警去冗余和遗漏证据补充来消除误报和漏报的影响, 保证了证据的完整性.

2 网络取证分析方法

本文进行网络取证分析的原始数据是通过分布在网络中的各个关键节点的入侵检测系统获得的, 利用开源入侵检测系统 Snort 分析捕获的数据包, 将得到的攻击报警保存在数据库中, 以备以后作为原始证据呈现给法庭. 通过对报警数据进行格式标准化处理、数据去冗余、报警关联、遗漏补充和报警聚合 5 个步骤, 最终得到入侵过程的证据链.

2.1 报警格式标准化

网络数据包经过 Snort 检测系统后生成报警 (Alert). 为了提高分析效率并简化报警信息, 本文对报警格式进行了精简, 只保留了分析所必需的信息. 根据 IDMEF 格式的定义, 选取以下 5 种报警属性:

1) 报警信息的 ID 号 Aid;

2) 报警发生的时间 time;

3) 报警类型 Alert_type;

4) 报警的源地址 source;

5) 报警的目的地址 dest.

约简后的报警格式为 Alert(Aid, Alert_type, source, dest, begin_time, end_time, Attack_class). 取证分析的目的是为了发现证据之间的联系. 根据标准化后的报警属性, 将报警信息之间的联系分为重复、聚类和关联 3 种类型.

定义 1 (重复关系) 两条报警 A_1 和 A_2 的 Alert_type、source 和 dest 值均相同, 且 (A_1.begin_time $-$ A_2.begin_time) $< \delta$ (δ 是预先设置的阈值).

定义 2 (聚类关系) 对于两条报警 A_1 和 A_2, 若

$$A_1.\text{source} = A_2.\text{source} \bigwedge A_1.\text{dest} = A_2.\text{dest} \bigwedge A_1.\text{Alert\_type} = A_2.\text{Alert\_type},$$

则称 A_1 与 A_2 满足聚类关系.

定义 3 (关联关系) 对于任意两个报警信息 A_1 和 A_2, 如果其 source 和 dest 的属性值相同, 且设 A_1 对应攻击的后果集合为 $\text{Con}(A_1)$, A_2 对应攻击的前提集合为 $\text{Pre}(A_2)$, 有 $\text{Con}(A_1) \subseteq \text{Pre}(A_2)$, 则称 A_1 和 A_2 存在关联关系, 记作 Correlation(A_1, A_2).

通过对报警信息进行因果关联分析, 挖掘出整个攻击的关联序列, 便能发现攻击者的意图, 从而给人们呈现出更加直观的犯罪证据. 为了更清晰地还原攻击的过程, 对报警类型进行划分, 定义如下:

定义 4(原始报警) 由入侵检测系统产生经格式标准化后产生的报警, 没有经过其他处理, 不可再分.

定义 5(元报警) 原始报警经过重复聚合后的报警.

定义 6(攻击场景) 将经过处理的报警按照因果关联的关系串联起来, 对应于一次入侵过程.

2.2 去冗余方法

去冗余是对报警的精简和合并, 为报警聚合和关联分析做准备. 主要包括: 1) 合并重复报警数据, 如对某一主机不同端口的扫描产生的重复报警; 2) 去除攻击失败的报警, 这些报警信息在分析整个攻击场景时是不需要的. 在分析报警数据之前去除这些冗余信息, 有助于提高攻击场景的清晰性和证据重构的效率.

合并重复报警时, 将具有相同源 IP 地址和相同目的 IP 地址以及相同报警类型的报警合并为元报警, 其报警时间落入一个自扩展的时间窗口. 当原始报警的时间超出元报警的时间窗口时, 元报警的时间窗口在预定义界限 T 内自动扩展. 这样能够把连续重复的原始报警合并为一个具有适当 T 时间窗口的元报警.

攻击失败的报警对于证据重构是没有意义的. 例如, 在缓冲区溢出攻击中, 攻击者首先进行缓冲区溢出尝试; 为了检测入侵是否成功, 攻击脚本尝试以新建的 root 用户通过 telnet 登录, 受害主机会发出相应

的报警. 若溢出成功, 受害主机发出 RSERVICES Rsh root 报警, 表示攻击者已经获得了远程 root 权限; 若不成功, 发出 INFO TELNET login incorrect 报警, 表示远程登录不成功. 在入侵场景重构中, 只需要保留攻击者成功登录系统的事件序列, 而登录不成功对应的事件序列则不需要保留.

2.3　场景重构

本文提出了一种基于规则推理和因果关联的场景重构方法, 以寻找攻击步骤之间的"因果关系"为入口, 在大量离散的单个警报中挖掘出具有时序性和逻辑性的报警序列, 并将其映射为同一攻击计划的不同步骤或不同阶段, 将逻辑相关的攻击步骤关联起来, 从而重现完整的证据链. 本文算法的基础是 Peng 等[6] 提出的基于前因后果的报警信息关联方法. 为了表示报警的前因和后果, 本文采用了超级报警的形式.

定义 7(超级报警)　超级报警 T 用一个三元组 (fact, prerequisite, consequence) 表示, 其中 prerequisite 表示报警发生的前提, consequence 表示攻击成功后所获得的信息, fact 表示本次报警事件. prerequisite, consequence 中所涉及的变量均存在于 fact 中.

定义 8(超级报警的实例)　对于一个超级报警 T, $P(T)$ 表示 T 的前提, $C(T)$ 表示 T 的结果, h 为 T 的实例, $P(h)$ 是 $P(T)$ 中的自由变元对应于 h 的具体值, 同样, $C(h)$ 是 $C(T)$ 中的自由变元对应于 h 的具体值. 实例 h 暗示其前提条件 prerequisite 必须为真, consequence 可能为真.

定义 9(报警间关联)　对于两个超级报警实例 h_1 和 h_2, 如果

$$C(h_1)\bigcap P(h_2)\neq \varnothing \bigwedge h_1.\text{end\_time} < h_2.\text{begin\_time},$$

则称 h_1 与 h_2 间存在因果关系, 可以把这两条报警关联起来.

本文将所有的因果关联关系都预先根据经验定义在知识库中. 由于攻击类型和手段的变化, 知识库需要不断更新和完善. 根据知识库里的因果关联规则, 将去冗余后的各条报警实例化, 存入报警信息表 CombinedAlert; 前提条件的相关信息存入表 PrereqSet, 结果状态的相关信息存入表 ConseqSet. 根据前提和结果条件进行因果关联, 并将报警的直接关联信息存储在 Correlate 表中, 前提条件不完全满足的孤立报警存储在 IsolatedAlert 表中. 根据 Correlate 表和 IsolatedAlert 表中的报警重构入侵场景, 并对遗漏报警进行补充.

本文的因果关联算法在文献 [6] 方法的基础上加入了时间约束, 即两条报警是有时间限制的, 若超过了一定的时间限制, 即使某报警的结果与另一条报警的前提条件相匹配, 两条报警也不能进行因果关联. 关联分析时, 要从时间约束、状态约束和参数约束 3 个方面考虑. 具体过程如算法 1 所示.

算法 1　因果关联算法.

Step 1: 取 PrereqSet 表中的每一条记录, 依次与 ConseqSet 表中的记录进行匹配;

Step 2: 若 ConseqSet 表中的第 j 条记录与 PrereqSet 表中的第 i 条记录匹配, 则匹配成功, 直接在 Correlate 表中插入一条记录, 该条记录的 PreMid 为 PrereqSet 表中对应的 Mid, ConMid 为 ConseqSet 表中对应的 Mid (本文设定的时间阈值 $T = 20$ min).

Step 3: 若匹配不成功, 则直接将该条记录插入 IsolatedAlert 表中.

2.4　遗漏证据补充方法

目前的关联算法大都是被动地发现报警间的关联, 没有主动挖掘攻击事件的规律. 本文在因果关联的基础上, 发掘事件间的间接因果关系, 重现犯罪的证据链. 根据约束条件排除不可能的遗漏攻击, 根据报警的类型确定是否误报.

对一个超级报警进行关联分析后, 如果通过直接关联没有找到与之相关联的报警, 则进行漏报关联分析, 补充遗漏报警, 重构攻击场景. 本文采用以下补全证据链的策略: 对于某一入侵场景, 从该场景中报警类型最高且时间最晚的报警开始, 向前补充; 补充完成后, 检查该入侵过程是否完整, 若不完整, 则继续进行遗漏报警补充; 将报警类型较低且为孤立节点的报警视为误报警, 不予关联和补充.

2.5　报警聚合算法

本文提出一种将相似度和抽象性相结合的报警聚合算法. 相似性聚合指的是把属于同一攻击步骤的报警根据报警的属性相似性结合在一起, 抽象性聚合指的是聚合某一攻击步骤的子步骤, 即聚合同一攻击步骤的不同报警. 属性的相似性又分为全局相似性和局部相似性, 全局相似性定义为各属性的相似度的加权平均. 聚合的最终结果是产生若干个聚合报警, 聚合报警代表了攻击场景中的一个可以描述的完整步骤. 该算法主要包括两个步骤: 1) 具体步骤映射为抽象步骤; 2) 不同场景的报警聚合.

报警聚合的方法有两种, 一种是将新报警与超级报警进行聚合, 另一种是将新报警与超级报警中的任意若干报警进行比较. 当超级报警的属性变化较大, 并且超级报警中的报警数量较多时, 进行计算消耗的时间过长, 会影响算法的效率. 因此采用最近的 5 条

报警与新报警进行比较, 取平均值作为最终的相似度.

算法 2 报警聚合算法.

Step 1: 首先提取各完整入侵场景中的元报警 $MA_1, MA_2, \cdots, MA_n$.

Step 2: 对于新报警 A, 寻找可以聚合的报警 Search(A, MA_i):

Step 2.1: 若 A.Alert_type $=$ MA_i.Alert_type, 则转 Step 2.2, 否则插入聚合队列;

Step 2.2: 若 IP 地址相似度 Sim_IP(A.source, MA_i.source) > IP 阈值, 则转 Step 2.4, 否则转 Step 2.3;

Step 2.3: 若 Sim_IP(A.dest, MA_i.dest) > IP 阈值, 则转 Step 2.6, 否则转 Step 2.4;

Step 2.4: 若 Sim_IP(A.dest, MA_i.source) > IP 阈值, 则将 A 的 dest 与 source 互换, 并转 Step 2.5, 否则插入聚合队列;

Step 2.5: 根据报警类别计算时间相似度, 若时间相似度 $\mathrm{Sim\_time}$ (A.begin_time, MA_i.begin_time) > time 阈值, 则转 Step 2.6, 否则插入队列;

Step 2.6: 若全局相似度 $\mathrm{Sim_{All}}(A, MA_i) >$ AllSim 最小阈值, 则转 Step 3, 否则直接插入聚合队列.

Step 3: 聚合报警 Aggregation Alert (A, MA_i):

Step 3.1: 生成一个新报警 N, 有

N.begin_time $=$

$\min\{MA_i$.begin_time, A.begin_time$\}$,

N.end_time $=$

$\max\{MA_i$.end_time, A.end_time$\}$;

Step 3.2: IP 地址取 A 和 MA_i 的 IP 地址的合并.

Step 4: 将各入侵场景中报警的 Mid 映射为对应的 Ag_id, 删除场景中重复的 (ConMid, PreMid).

Step 5: 比较各入侵场景是否相同, 将相同的入侵场景合并.

2.6 证据的完整性及司法有效性

经过场景重构和报警聚合, 在得到入侵场景的同时, 与入侵场景相关的报警信息也成为重要的电子证据. 为了保证电子证据的完整性和司法有效性, 需要对其进行相应的保全处理. 目前在静态取证的研究中, 已经总结出一套比较成熟的证据保全方法, 得到了司法部门的认可. 而网络取证得到的证据, 也可以借鉴这些方法. 例如, 可以利用单向哈希函数计算证据数据的数字摘要, 与原始证据一起保存.

2.7 证据链的完整性

入侵检测系统的准确率达不到 100%, 必然存在误报和漏报. 本文方法是通过去冗余技术和报警补充策略, 尽量减少误报和漏报的影响. 但是, 仅仅依靠入侵检测系统的报警有时难以得到完整的证据链, 这时便需要综合其他方面的证据 (如系统日志、主机上的其他证据等). 同时, 关于证据链的完整性还没有公认的度量标准, 需要结合具体案件由相关专家进行判定.

3 实验结果及分析

3.1 实验数据

本文用到的入侵检测系统为 Snort 2.9.2.2, 实验中采用的实验数据集是美国麻省理工学院林肯实验室提供的 DAPPA 2000 入侵检测评估数据流 LLDOS. 该数据集包括 LLDOS 1.0 和 LLDOS 2.0 两个攻击场景实例, 本实验中用到的是 LLDOS 1.0. 在 LLDOS 1.0 攻击场景中, 攻击者通过 Solaris sadmind 服务漏洞攻陷并控制了 Eyrie 空军基地网络中的 3 台主机, 上传了 Mstream 工具, 并对一个政府网站发动了 DDoS 攻击.

3.2 实验结果

在 Linux 系统下利用 Tcpreplay 网络流量重放工具向 Snort 检测的网段重放数据, 当 Snort 检测到攻击数据流时便会产生报警数据. Snort 产生的原始报警共 5 654 条, 按照 IDMEF 格式经标准化后存储在 Original Alert 表中. 通过去冗余算法共去除失败报警 148 个; 报警合并后产生的元报警共 1 049 条, 存储在 CombinedAlert 中.

根据关联结果和遗漏补充得到的 3 个入侵场景如下:

Attack scenario 1

ICMP PING → RPC portmap sadmind request → RPC sadmind UDP PING → RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt → RSERVICES Rsh root → Rsh host command → mstream_zomnie → DDoS

RPC portmap sadmind request → RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt

RSERVICES Rsh root → mstream_zomnie

数据包信息

323 324 325 → 754 755 → 756 757 → 1824 → 2279 2280 2281 2282 2283 2284 → Rsh host command → 3354 → DDoS

754 755 → 1824

2279 2280 2281 2282 2283 2284 → 3354

Attack scenario 2

ICMP PING → RPC portmap sadmind request → RPC sadmind UDP PING → RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt → RSERVICES Rsh root → Rsh host

command → mstream_zomnie → DDoS

RPC portmap sadmind request → RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt

RSERVICES Rsh root → mstream_zomnie

数据包信息

355 356 357 358 → 923 924 → 925 926 → 1830 → 2290 2291 2292 2293 → Rsh host command → 3356 → DDoS

923 924 → 1830

2290 2291 2292 2293 → 3356

Attack scenario 3

ICMP PING → RPC portmap sadmind request → RPC sadmind UDP PING → RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt → RSERVICES Rsh root → Rsh host command → mstream_zomnie → DDoS

RPC portmap sadmind request → RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt

RSERVICES Rsh root → mstream_zomnie

数据包信息

388 389 325 → 957 958 → 959 960 → RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt → 2297 2298 2299 2300 → Rsh host command → 3370 3371 → DDoS

957 958 → RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt

2297 2298 2299 2300 → 3370 3371

Attack scenario 对应的是报警类型的关联图, 数据包信息为场景对应的原始数据包, 若在 Original Alert 中没有找到相应的 Aid, 证明该报警遗漏, 则用报警类型 Alert_type 代替. 根据报警聚合结果, 合并入侵场景中重复元组, 最后得到的证据链如图 1 所示.

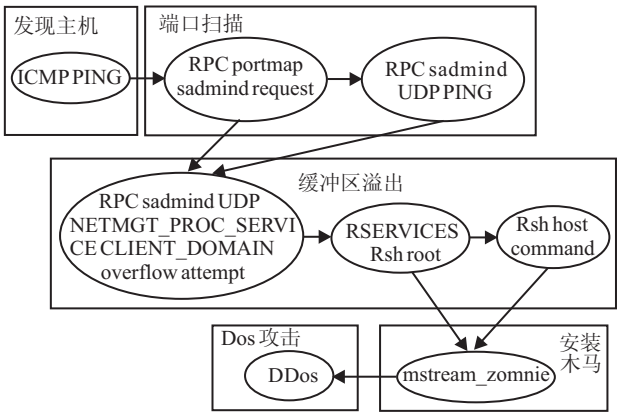

图 1　完整证据链

3.3　对比分析

由场景重构结果可知, 一台攻击主机 (IP 地址为 202.77.162.213) 通过缓冲区溢出成功占领了 3 台受害主机 (IP 地址分别为172.16.115.20、172.16.112.10 和 172.16.112.50). 其中, 3 台受害主机遗漏了报警 Rsh host command 和 DDoS, 导致了证据链的分裂. Rsh host command 是入侵者通过 Rsh 服务运行 Agent 的关键步骤, 而 DDoS 是攻击的最终结果. 由此可知, 这两个报警对于证据链的重构很重要, 必须通过遗漏补充来补充这两种报警. 此外, 对于受害主机还遗漏了报警 RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_ DOMAIN overflow attempt. 本文通过直接和间接的关联分析, 将整个攻击的步骤完整地显现出来.

即使某些攻击只出现过一次报警, 本文提出的方法也可以通过分析取得相应的证据, 因此适用于实时取证. 本文方法对漏报进行了处理, 对于失败的或者无用的报警进行了去除, 保证了证据的完整性; 同时通过报警聚合将证据链进行了合并, 形成了抽象和具体两条证据链, 分别对应于入侵者和受害者, 保证了证据链的清晰性; 本文方法不完全依赖于已知的攻击模型, 也能发现未知的攻击场景. 同其他的网络取证分析方法相比, 本文提出的方法在入侵事件的重构方面具有明显的优点, 对比分析如表 1 所示.

表 1　本文方法与相关工作的对比分析

方 法	实时取证	漏处理报	证据正确性	证据链的完整性	证据链针对者	发现未知攻击场景
因果关联	√	×	×	×	受害者	√
序列挖掘	×	×	×	×	攻击者,受害者	√
报警聚类	×	×	×	×	攻击者	√
已知攻击模型	√	√	×	√	攻击者, 受害者	×
本文方法	√	√	√	√	攻击者,受害者	√

4　结　论

在网络取证的研究中, 报警聚合能有效抽象出不同主机的行为, 场景重构对每台主机可以实现证据链的重现. 本文把两者结合起来, 提出了基于报警聚合和场景重构的网络取证分析方法: 先进行事件重构再进行报警聚合, 既保证了证据的准确性又保证了证据的全面性; 通过去冗余和遗漏报警补充策略, 减少了误报和漏报对取证结果的影响; 在报警聚合过程中,

增加了聚合相同攻击步骤的不同报警的思想, 有利于从宏观上了解入侵者的入侵动机和过程, 提高了证据链的清晰性. 在下一步工作中, 将对提出的算法进行进一步优化, 并对入侵知识库进行完善.

参考文献(References)

[1] 丁丽萍. 基于网络数据流的计算机取证技术[J]. 信息网络安全, 2005(6): 74-76.
(Ding L P. Computer forensics technologies based on network traffics[J]. Netinfo Security, 2005(6): 74-76.)

[2] Curran K, Morrissey C, Fagan C, et al. Monitoring hacker activity with a honeynet[J]. Int J of Network Management, 2005, 15(2): 123-134.

[3] Valdes A, Skinner K. Probabilistic alert correlation[C]. Proc of Int Conf on Recent Advances in Intrusion Detection (RAID'2001). Berlin: Springer, 2001: 54-68.

[4] Xu D, Peng N. Alert correlation through triggering events and common resources[C]. Proc of ACSAC'2004. Washington DC: IEEE Computer Society Press, 2004: 360-369.

[5] Zhou J, Heckman M, Reynolds B, et al. Modeling network intrusion detection alerts for correlation[J]. ACM Trans on Information and System Security, 2007, 10(1): 1-13.

[6] Peng N, Yun C, Douglas S. Techniques and tools for analyzing intrusion alerts[J]. ACM Trans on Information and System Security, 2004, 7(2): 274-318.

[7] Qin X, Lee W. Statistical causality analysis of INFOSEC alert data[C]. Proc of RAID'2003. Berlin: Springer, 2003: 73-93.

[8] Noel S, Robertson E, Jajodia S. Correlating intrusion events and building attack scenarios through attack graph distances[C]. Proc of the 20th Annual Computer Security Applications Conf. Washington DC: IEEE Computer Society Press, 2004: 350-359.

[9] Debar H, Wespi A. Aggregation and correlation of intrusion-detection alerts[C]. Proc of the 4th Int Symposium on Recent Advances in Intrusion Detection. Berlin: Springer, 2001: 85-103.

[10] Locatelli F E, Gaspary L P, Melchiors C. Spotting intrusion scenarios from firewall logs through a case-based reasoning approach[C]. Lecture Notes in Computer Science. Berlin: Springer, 2004, 3278: 196-207.

[11] Goldman R P, Heimerdinger W. Information modeling for intrusion report aggregation[C]. Proc of DISCEX'01. Washington DC: IEEE Computer Society Press, 2001: 329-343.

[12] Treinen J J, Thurimella R. A framework for the application of association rule mining in large intrusion detection infrastructures[C]. Proc of the 9th Int Conf on Recent Advances in Intrusion Detection. Berlin: Springer, 2006: 1-18.

[13] 张有东, 曾庆凯, 王建东. 网络协同取证计算研究[J]. 计算机学报, 2010, 33(3): 504-513.
(Zhang Y D, Zeng Q K, Wang J D. Studies of network coordinative forensics computing[J]. Chinese J of Computers, 2010, 33(3): 504-513.)

(责任编辑: 孙艺红)